

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A computer program product, tangibly embodied in ~~an information carrier~~ a storage device, the computer program product being operable to cause data processing apparatus to perform operations comprising:

receiving an original design-time representation of an application, the original design-time representation for use in a first run-time environment for executing applications having been developed in a first design-time environment, the first design-time environment using a first programming model comprising one or more first model elements including screens and processing logic for each screen, the original design-time representation including one or more application screens and original processing logic for each application screen; and

generating a converted design-time representation of the application based on the original design-time representation, the converted design-time representation for use in a second run-time environment for executing applications having been developed in a second design-time environment, the second design-time environment using a second programming model comprising one or more second model elements including models, views, and controllers, the converted design-time representation including one or more application views based on the one or more application screens, and converted processing logic based on the original processing logic, the converted processing logic capable of being executed in the second run-time environment.

2. (Original) The computer program product of claim 1, wherein the first programming model is the SAP Dynpro programming model, and the second programming model is the SAP Web Dynpro programming model.

3. (Original) The computer program product of claim 1, wherein generating a converted design-time representation of the application comprises:
converting each application screen to a corresponding application view; and
converting the original processing logic for each application screen to the converted processing logic.

4. (Original) The computer program product of claim 3, wherein:
each application screen comprises one or more controls from a first set of controls defined in the first programming model;
the second programming model defines a second set of controls; and converting each application screen comprises selecting a corresponding control from the second set of controls for each control in the application screen.

5. (Original) The computer program product of claim 4, wherein each control comprises an attribute, and wherein converting each application screen further comprises, for each control in the application screen, setting the attribute of the corresponding control to match the attribute of the control in the application screen.

6. (Original) The computer program product of claim 3, wherein the original processing logic comprises state control logic and one or more calls to one or more run-time modules in the first run-time environment, and wherein converting the original processing logic comprises:

generating corresponding state control logic that is executable by an adapter in the second run-time environment, the adapter being operable to interface with the run-time modules in the first run-time environment; and

converting the calls to the run-time modules into instructions to the adapter for invoking the run-time modules.

7. (Original) The computer program product of claim 3, wherein converting the original processing logic comprises generating one or more instructions to an adapter in the second run-time environment to perform a function not performed by the original processing logic.

8. (Original) The computer program product of claim 3, wherein:

converting the original processing logic comprises generating code to invoke an adapter in the second run-time environment; and

the code to invoke the adapter is formatted to resemble the original processing logic.

9. (Original) The computer program product of claim 8, wherein the code to invoke the adapter comprises one or more macros.

10. (Original) A system comprising:

a first run-time environment operable to execute run-time code generated from design-time representations of applications developed in a first design-time environment, the first design-time environment using a first programming model comprising one or more first model elements including models, views, and controllers;

a conversion module operable to:

receive an original design-time representation of an application, the original design-time representation for use in a second run-time environment for executing applications having been developed in a second design-time environment, the second design-time environment using a second programming model comprising one or more second model elements including screens and processing logic for each screen, the original design-time representation including one or more application screens and original processing logic for each application screen, the original processing logic including a call to a run-time module in the second run-time environment; and

generate a converted design-time representation of the application based on the original design-time representation, the converted design-time representation for use in the first run-time environment, the converted design-time representation including one or more application views based on the one or more application screens, and converted processing logic based on the original processing logic, the converted processing logic capable of being executed in the first run-time environment; and

an adapter operable to interface with the run-time module in the second run-time environment.

11. (Original) The system of claim 10, wherein the converted processing logic comprises an instruction to the adapter to invoke the run-time module based on the call to the run-time module in the original processing logic.

12. (Original) The system of claim 10, wherein:
the first programming model defines a first set of controls;
the second programming model defines a second set of controls; and
the converted design-time representation of the application comprises a corresponding control from the first set of controls for each control in the original design-time representation of the application.

13. (Original) The system of claim 10, wherein the converted processing logic comprises instructions that are formatted to resemble the original processing logic.

14. (Original) The system of claim 10, wherein the converted design-time representation of the application comprises additional processing logic not included in the original processing logic.

15. (Original) An apparatus comprising:

means for receiving an original design-time representation of an application, the original design-time representation for use in a first run-time environment for executing applications having been developed in a first design-time environment, the first design-time environment using a first programming model comprising one or more first model elements including screens and processing logic for each screen, the original design-time representation including one or more application screens and original processing logic for each application screen; and

means for generating a converted design-time representation of the application based on the original design-time representation, the converted design-time representation for use in a second run-time environment for executing applications having been developed in a second design-time environment, the second design-time environment using a second programming model comprising one or more second model elements including models, views, and controllers, the converted design-time representation including one or more application views based on the one or more application screens, and converted processing logic based on the original processing logic, the converted processing logic capable of being executed in the second run-time environment.

16. (Original) The apparatus of claim 15, wherein the first programming model is the SAP Dynpro programming model, and the second programming model is the SAP Web Dynpro programming model.

17. (Original) The apparatus of claim 15, wherein the means for generating a converted design-time representation of the application comprises:

means for converting each application screen to a corresponding application view; and

means for converting the original processing logic for each application screen to the converted processing logic.

18. (Original) A method comprising:

receiving an original design-time representation of an application, the original design-time representation for use in a first run-time environment for executing applications having been developed in a first design-time environment, the first design-time environment using a first programming model comprising one or more first model elements including screens and processing logic for each screen, the original design-time representation including one or more application screens and original processing logic for each application screen; and

generating a converted design-time representation of the application based on the original design-time representation, the converted design-time representation for use in a second run-time environment for executing applications having been developed in a second design-time environment, the second design-time environment using a second programming model comprising one or more second model elements including models, views, and controllers, the converted design-time representation including one or more application views based on the one or more application screens, and converted

processing logic based on the original processing logic, the converted processing logic capable of being executed in the second run-time environment.

19. (Original) The method of claim 18, wherein the first programming model is the SAP Dynpro programming model, and the second programming model is the SAP Web Dynpro programming model.

20. (Original) The method of claim 18, wherein generating a converted design-time representation of the application comprises:

converting each application screen to a corresponding application view; and
converting the original processing logic for each application screen to the converted processing logic.